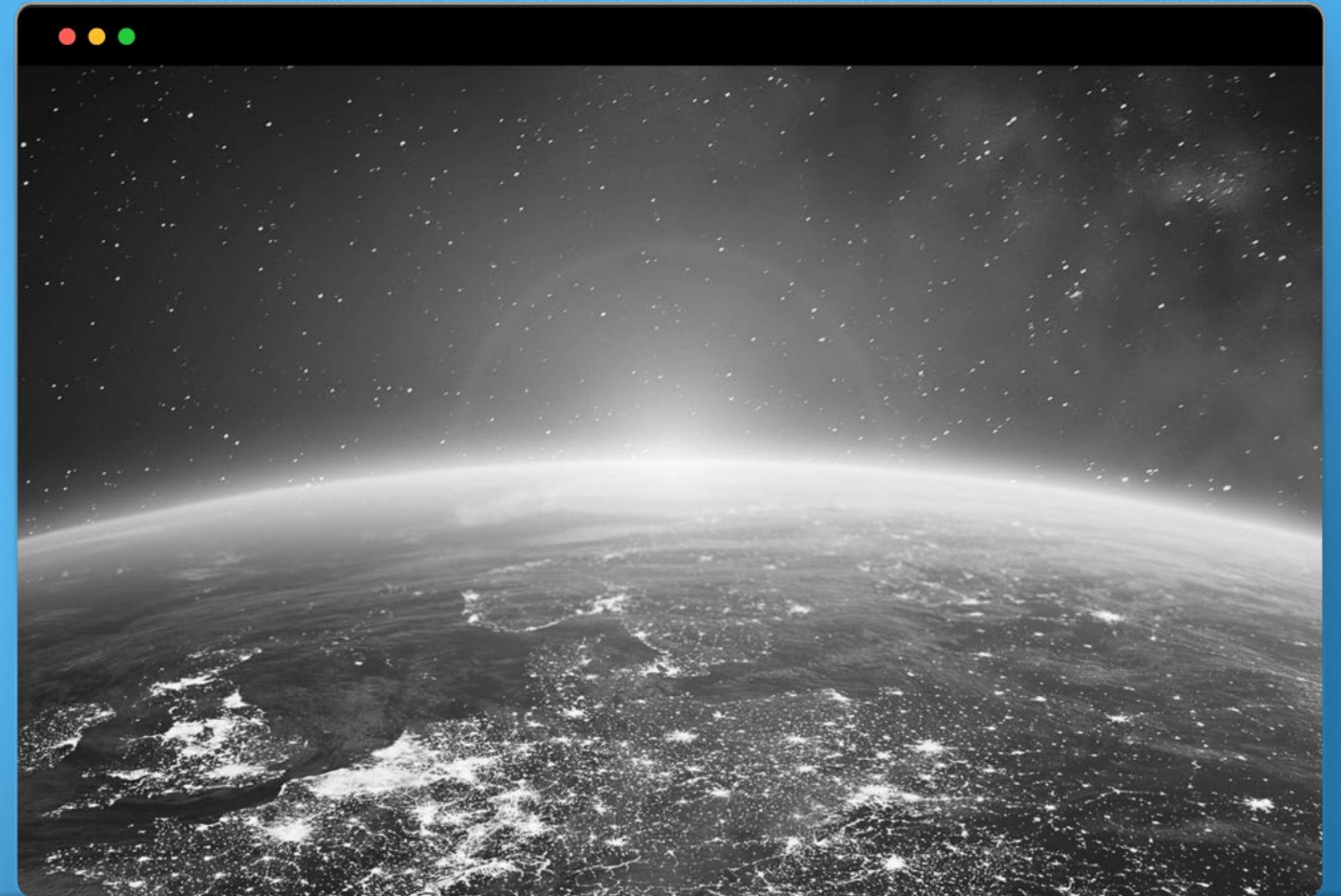


[CASE STUDY]



Full sp(r)eed ahead:
an operationally efficient
payment infrastructure





Why Spreedly migrated their payments platform from a NoSQL database to CockroachDB

INDUSTRY :

Fintech

CHALLENGES :

Depreciating legacy systems, moving to the cloud, and increasing operational efficiency .

SOLUTION :

A secure, scalable, resilient, cloud-native payment platform built on CockroachDB

About Spreedly

If you want to survive in the fintech industry, you have to keep up with customer demands. Think: automation, advanced security, submillisecond response times, API-driven requests. However, these demands can create a lot of operational complexity if you don't have the right technologies in place.

Founded in 2007, [Spreedly](#) is a payment orchestration company that has evolved its infrastructure to keep up with these ever-changing demands to deliver the best customer experience. They work with 100+ different currencies, serve 50K+ businesses (including merchants, aggregators, and fintechs), partner with 200+ services (including gateways, PSPs, fraud tools, and more), and securely store more than 1 billion payment methods.

Spreedly's platform is a cloud-based credit card vault that allows its customers to work with one or multiple payment gateways over time or simultaneously. All of this connectivity generates a lot of transactions — more specifically, monetary transactions (as opposed to when a transaction is considered a discrete body or unit of work).

Their cloud-based platform needed a scalable, resilient database that wouldn't add complexity to the application layer and allow them to "roll with the punches" that the fintech industry delivers — which is why they're building on CockroachDB.

“ ”

“CockroachDB is a good fit for Spreedly because it's really leaning into the future. There's a lot of features that work out of the box that older technologies cannot provide and it has allowed us to bring several legacy systems into a single system.”

– **Scott Traver**

Staff Engineer on the Application Architecture Team, Spreedly



Paying the price

In general, there are a lot of complications associated with securing financial data — from adhering to compliance regulations, to ensuring you never lose data, to guaranteeing consistent transactions, and much more. If you don't have the right infrastructure and security in place, it can become an operational nightmare.

Spredly originally built their platform on a NoSQL key-value data store. This database didn't have relationships, so they had the secondary index and secondary systems split from the primary application database. This setup became a struggle to maintain and wasn't performant; they needed to consolidate their data storage.

Additionally, Spredly wanted to move from on-premises datacenters to the public cloud. Making this transition with their NoSQL database setup would be difficult (considering all the separate systems) and they wanted a solution that could scale with them in the future.

They had a few requirements for a new database:

- High availability and enterprise security to protect customer data
- Ability to scale out to multiple regions and deliver low latency for a global audience
- Built for transactional workloads and guarantees data correctness
- Backed by a company with progressive mindset leaning into cloud-native approach

After researching for a new solution, they came across CockroachDB — a cloud-native, distributed SQL database designed to handle transactional workloads generated by global applications. CockroachDB met all of their requirements for a modern database solution and they felt confident in its ability to grow with them in the future.

100+

CURRENCIES

1 billion

PAYMENT METHODS

0

TIMES KILLED COCKROACHDB



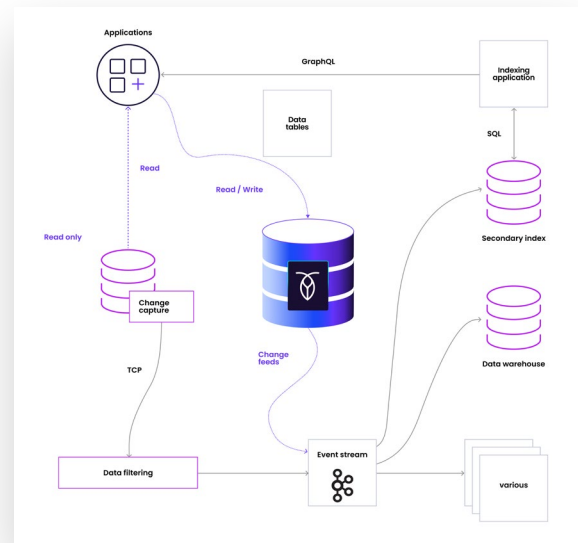
Right on the money

With the migration, Spreadly's main goal was to take their self-built split system — consisting of the database layer, the analysis layer, and secondary index layer — and consolidate all of those with CockroachDB. This new setup would simplify their architecture and allow them to move towards a cloud-first data storage approach.

The Spreadly team quickly realized that CockroachDB has features legacy technologies don't provide. For example, it has out-of-the box alerting and monitoring, the ability to partition data by locations, and change data capture (CDC) features. As they planned to consolidate systems — integrating the data storage layer with the analytics layer — they wanted to be as minimally disruptive as possible to downstream engineering and analytics teams. CDC is really paramount to that process. CDC helps data flow into the appropriate channels for analytics and get that analytics data fed back into the product to complete the data lifecycle. For Spreadly, analyzing their data is equally important to utilizing the data within the application.

Since Spreadly's platform is handling sensitive customer data, they embarked on a slow and safe migration process. Over time, they slowly throttled user data over to CockroachDB as they depreciated legacy systems. The diagram presents an overview of what their setup looked like during the migration.

ARCHITECTURE DIAGRAM Spreadly's architecture during migration



“The migration process has gone on without a hitch. CockroachDB is extremely resilient and has been much easier to manage. It provides monitoring and alerting, rolling upgrades and other features that have lowered maintenance. We’ve actually upgraded several times through the process and it’s been seamless.”

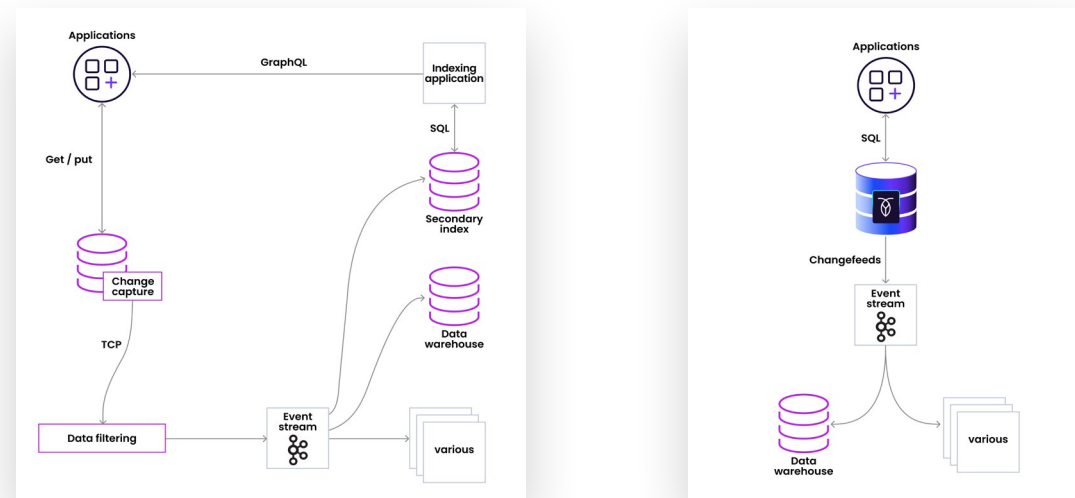
— **Scott Traver**, Staff Engineer on the Application Architecture Team, Spreadly

Cutting costs, not corners

Spredly is now in production with CockroachDB which they are self-hosting on AWS across multiple availability zones. They are storing a lot of transactional data in CockroachDB: customer requests from different providers / gateways / fraud vendors / network tokenization providers, configuration objects used to operate the platform, customer connectivity to third parties and gateways, and so on.

Here's a high-level view of what their architecture looked like before and after the migration.

ARCHITECTURE DIAGRAM
Spredly's simplified



As you can see from the diagram, they were able to depreciate several systems (3 custom code bases, 2 databases, 3 protocols) and consolidate on CockroachDB. Now they have a cloud-first data storage approach which will allow them to expand to multiple cloud regions in the future. With a multi-region setup, they will be able to provide low-latency access (by leveraging data localization capabilities) for their customers spanning geographies.

Throughout the migration process, they remarked that they were impressed with the high level of support from the Cockroach Labs team and the thorough documentation. Scott Travers, an engineer who manages CockroachDB, said that, "in terms of setting up CockroachDB and operating it, it's really just been a piece of cake."

While operations have been quite smooth, Scott mentioned that his team spent some time training colleagues on CockroachDB's more modern features. It's important that the app engineers and R&D teams understand the intricacies of an advanced system and the new capabilities it provides.

“ ”

“Transaction data is very important to us because we want our customers to have the most up-to-date record of their activity. We store that data and expose it and the correctness of that data in almost real time is extremely important to our customers.”

– **Scott Traver**

Staff Engineer on the Application
Architecture Team, Spredly

What's Next

Now that they've been able to retire several legacy systems, their operations teams are much more efficient. In terms of where the business is headed in the future, for Spreadly the "customers drive the opportunity." Where there's customer demand, Spreadly and CockroachDB will follow!

[Payments Platform Architecture Challenges | Spreadly & CockroachDB](#)

Ready to get started?

Spin up your first cluster in just a few clicks.

“ ”

“When I first learned the migration to CockroachDB was my responsibility, I set it up on my local machine and I tried to kill it. I tried to knock it over, and I thought my laptop was malfunctioning. It started to heat up and the fan was whirling like a jet engine, and I couldn't knock it over. I think CockroachDB is a really appropriate name for a technology that is so resilient.”

– **Scott Traver**

Staff Engineer on the Application
Architecture Team, Spreadly

